

Python: module vcs.fillarea

vcs.fillarea

[index](#)

Fillarea (Tf) module

Modules

[vcs.Canvas](#)

[Numeric](#)

[vcs.VCS_validation_functions](#) [vcs.vcs](#)

Classes

Tf

class *Tf*

Class: Tf

Fillarea

Description of Tf Class:

The Fillarea class object allows the user to edit fillarea attributes: fillarea interior style, style index, and color index.

This class is used to define an fillarea table entry used in VCS. It can be used to change some or all of the fillarea attributes in an existing fillarea table entry.

Other Useful Functions:

`a=vcs.init()`

Constructor

`a.show('fillarea')`

Show predefined fillarea

`a.update()`

Updates the VCS Canvas at

`a.mode=1, or 0`

If 1, then automatic update

0, then use update function

update the VCS Canvas.

Example of Use:

`a=vcs.init()`

To Create a new instance of fillarea use:

`fa=a.createfillarea('new','def37')` # Copies content of 'def37'

`fa=a.createfillarea('new')` # Copies content of 'default'

To Modify an existing fillarea use:

`fa=a.getfillarea('red')`

`fa.list()`

Will list all the fillarea

```

There are three possibilities for setting the isofill style (Ex)
fa.style = 'solid'
fa.style = 'hatch'
fa.style = 'pattern'
fa.index=1 # Range from 1 to 20
fa.color=100 # Range from 1 to 256

```

Specify the fillarea index:

```

fa.index=1
fa.index=2
fa.index=3
fa.index=4
fa.index=5
fa.index=6
fa.index=7
fa.index=8
fa.index=9
fa.index=10
fa.index=11
fa.index=12
fa.index=13
fa.index=14
fa.index=15
fa.index=16
fa.index=17
fa.index=18
fa.index=19
fa.index=20

```

```

fa.priority=1 # Set the graphics priority
fa.viewport=[0, 1.0, 0,1.0] # FloatType [0,1]x[0,1]
fa.worldcoordinate=[0,1.0,0,1.0] # FloatType [#,#]x[#,#]

fa.x=[[0, .1, .2], [.3, .4, .5]] # List of FloatTypes
fa.y=[[.5, .4, .3], [.2, .1, 0]] # List of FloatTypes

```

Methods defined here:

```

__init__(self, parent, Tf_name=None, Tf_name_src='default', createTf=0)
#####
#
# Initialize the fillarea attributes.
#
#####

__setattr__(self, name, value)
#####
#
# Set fillarea attributes.
#
#####

```

```

list(self)
#####
#
# Fillarea out line members (attributes).
#
#####

script(self, script_filename=None, mode=None)
Function:          script                                # Calls _vcs.s

Description of Function:
    Saves out a fillarea graphics method in Python or VCS s
    designated file.

Example of Use:
    script(scriptfile_name)
        where: scriptfile_name is the output name of the
              mode is either "w" for replace or "a" for

    Note: If the the filename has a ".py" at the end
          Python script. If the filename has a ".scr
          produce a VCS script. If neither extension
          default a Python script will be produced.

a=vcs.init()
fa=a.createfillarea('temp')
fa.script('filename.py')           # Append to a Python file
fa.script('filename.scr')         # Append to a VCS file "f
fa.script('filename','w')         # Create or overwrite to

```

Functions

```

getTfmember(self, member)
#####
#
# Function:          getTfmember
#
# Description of Function:
#     Private function that retrieves the fillarea members from
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
#     getTfmember(self,name)
#         where: self is the class (e.g., If)
#               name is the name of the member that is being
#
#####

```

```

renameTf(self, old_name, new_name)
#####
#
# Function:      renameTf
#
# Description of Function:
#     Private function that renames the name of an existing fill
#     graphics method.
#
#
# Example of Use:
#     renameTf(old_name, new_name)
#         where: old_name is the current name of fillarea gra
#         new_name is the new name for the fillarea gra
#
#####

```

```

setTfmember(self, member, value)
#####
#
# Function:      setTfmember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:
#     setTfmember(self,name,value)
#         where: self is the class (e.g., Tf)
#         name is the name of the member that is being
#         value is the new value of the member (or att
#
#####

```